

Create PT Prep

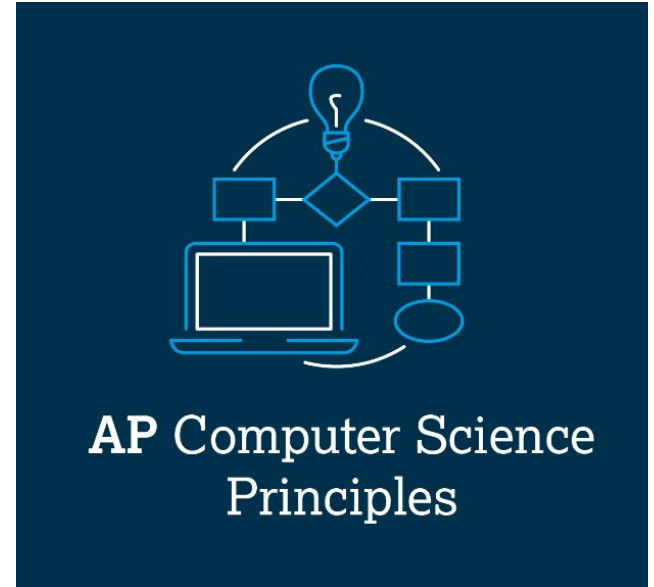
Selecting a project to meet the requirements of the Create Performance Task



AP CSP Create Performance Task

Things to know about the Create PT:

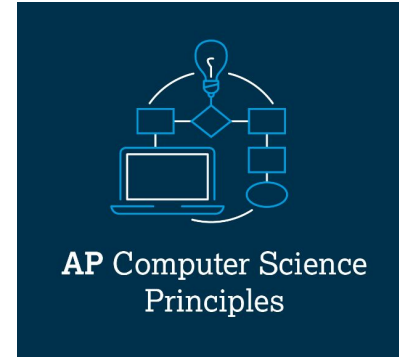
- You can work with a partner
- Your teacher cannot help you
- You cannot use an assigned program or Practice PT for your project, but it can be similar
- Your classmates can help you
- You will be given 9 hours of class time to work on the project



AP CSP Create Performance Task

Create PT specific requirements:

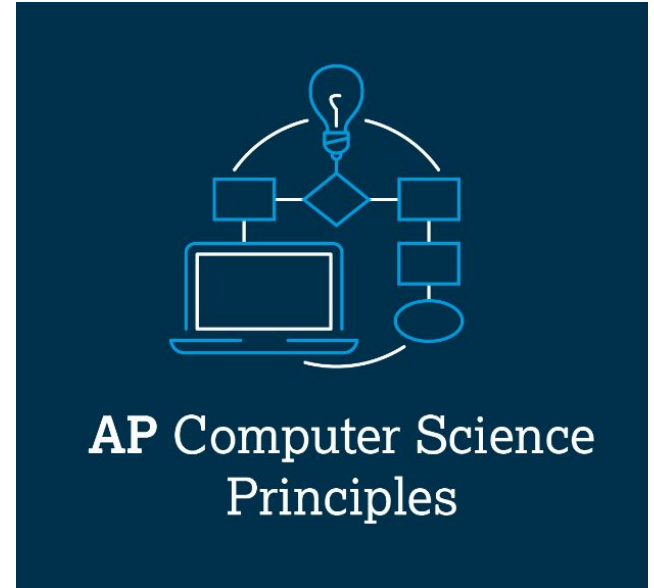
- Has input (button presses)
- Has output (screen, pixels, audio, etc.)
- Creates a list
- Uses a list in a meaningful way
- Has a function with a parameter
 - Parameter has an effect on the functionality of the procedure (Parameter used in if statement)
- Function has:
 - Selection (If statement)
 - Iteration (Loop)



AP CSP Create Performance Task

For this lesson, you will focus on selecting a project

- Ideas for projects that meet the requirements



Selecting a project

Create Performance Task



Selecting a project

Throughout the PT practice lessons, you created several projects that meet, or can meet, the requirements for the Create Performance Task

- Practice #2
- Practice #3
- Practice #4
- Practice #5
- Practice #6



Selecting a project

Let's look at each project and develop ideas for a new project.

- Remember: You CAN NOT submit a project that you completed for an assignment.
- So you cannot use any of your practice projects, missions, or remix projects.
- Your Create PT must be new and original.
- However, your Create PT can be similar, or based on, any of these projects.



Practice #2

Look at the project from Create PT Practice #2.

- This project gave the user a choice between two topics, and then displayed information from lists about the topic
- The user could scroll, do a slideshow, or random item from the list
- This project used two lists for each choice, but you could easily do something similar with one list for each choice

```
from codex import *
from time import sleep
import random

dbacks_pos = ["pitcher", "catcher", "1st Base",
              "2nd Base", "3rd Base", "shortstop"]
dbacks_players = ["Merrill Kelley", "Gabriel Moreno", "Cody
                  "Ketel Marte", "Evan Longoria", "Gerald
rangers_pos = ["cather", "3rd base", "1st base",
               "shortstop", "2nd base", "outfielder"]
rangers_players = ["Mitch Garver", "Josh Jung", "Nathani
                  "Corey Seager", "Marcus Semian", "Adolis
```

```
def intro():
    display.print("Welcome to the ")
    display.print("World Series")
    display.print("A = Diamondbacks")
    display.print("B = Rangers")
    display.print("")
    display.print("R = Scroll forward")
    display.print("L = Slideshow")
    display.print("U = Random player")
    display.print("D = Quit")
```



New projects similar to Practice #2

Give the user a choice between two things:

- Have one topic and two different aspects of it
 - One list of images and another list of sound clips
 - One list of images, and the other list of facts, or dates, etc.
 - Two different lists about one topic – like album covers and band tours
- Use something other than `display.print()` statements to display the information
- Make the project like study cards
 - Choose between two different subjects to review
 - Have one list be questions (think of the answer), and the other list be answers (think of the question)



Practice #3

Look at the project from Create PT Practice #3.

- This project gave the user a choice between easy and hard
- The user played a game of pressing the correct button after a prompt
- This project used two lists: one for the instructions and one for the buttons to press in sequence

```
messages = ["Press Up", "Press Down", "Press Left", "Press Right"]
btns = [BTN_U, BTN_D, BTN_L, BTN_R]

count = 0

# One function for game play
def play_game(choice):
    global count
    if choice == 1:
        delay = 1.5
    else:
        delay = 0.75
    for index in range(len(messages)):
        message = messages[index]
        btn = btns[index]
        display.show(message)
        sleep(delay)

    pressed = buttons.is_pressed(btn)
    if pressed:
        pixels.set(index, GREEN)
        count = count + 1
    else:
        pixels.set(index, RED)
```



New projects similar to Practice #3

Give the user a choice between two versions of a game:

- Add another level: mild, medium and spicy
 - Use more buttons, or different delays
- Use a color to indicate which button to press
 - Use the pixels for the color – set all pixels the same color
 - Use the screen for the color – `display.fill(color)`
- Use arrows or other images to point to the button to press instead of pixels
 - The easy version could use four buttons, and the hard version could use all six buttons



Practice #4

Look at the project from Create PT Practice #4.

- This project is the same as Practice #3, but added a counter
- The user played a game of pressing the correct button after a prompt
- The counter was used as a parameter for an ending

```
def ending(count):  
    # turn off all pixels and clear screen  
    pixels.set([BLACK, BLACK, BLACK, BLACK])  
    display.clear()  
  
    if count == len(btns):  
        end_message = "You won!"  
        col = GREEN  
    elif count == 0:  
        end_message = "You lost"  
        col = RED  
    else:  
        end_message = "Keep trying"  
        col = BLUE  
  
    display.draw_text(end_message, scale=3, x=30, y=30)  
    # running pixel lights  
    for num in range(30):  
        pixels.set(num%4, col)  
        sleep(0.2)  
        pixels.set(num%4, BLACK)
```



New projects similar to Practice #4

The user plays a game, and the project keeps score:

- Pick a different game and add a counter
 - Reaction time is a game. Do something similar and add a counter.
 - What other projects could you make into a game? And then add a counter.
- Use something that isn't a game, but use a counter
 - Something like billboard – keep track of how many times a particular item is chosen and display when the user quits
 - Something like answerbot – divide the sayings into positive and negative; use two counters to keep track of how many positive and negative messages were given and display when the user quits



Practice #5

Look at the project from Create PT Practice #5.

- This project combined two missions into one program
- User chose which project
- At least one of the projects must have a list and a loop

```
...
Create PT Practice #5
combine two missions, one with a list and loop
to meet requirements of Create PT
...

from codex import *
from time import sleep

my_list = ["Ahoy", GREEN, pics.HAPPY, pics.SAD, F
           pics.ASLEEP, BLUE, pics.HEART, pics.TIA
           ]
choice = 0
LAST_INDEX = len(my_list) - 1
delay = 1.0
my_choice = 0

# function with parameter and loop
def do_the_choice(my_choice):
    global delay, choice
    if my_choice == 1:
        # heart beat intro
        while True:
```

```
def intro():
    global my_choice
    display.clear
    display.print("Press A-heartbeat")
    display.print("Press B-billboard")
    display.print("Press D-quit")
    my_choice = 0
```



New projects similar to Practice #5

The user chooses between two projects:

- Pick a two different missions
 - Make changes to the missions so they aren't the exact same as the ones you turned in for an assignment
 - At least one must use a list and a loop
- Create a game spinner that goes both directions
 - Create a list for the arrows
 - One choice the spinner goes clockwise
 - One choice the spinner goes counter-clockwise
 - Keep all the code in one function!
- Create a project that can be used for games
 - Create a list for the arrows and use the game spinner
 - For the second option, create a dice roll



Practice #6

Look at the project from Create PT Practice #6.

- This project played the memory game with Simon
- It uses a list to keep track of the sequence
- It doesn't yet include a function with a parameter, if and loop

```
def simon_turn():
    display.clear()
    display.draw_text("Simon's turn", x=20, y=100)
    pixel = random.randrange(4)
    sequence.append(pixel)
    for item in sequence:
        pixels.set(item, GREEN)
        sleep(delay)
        pixels.set(item, BLACK)
        sleep(delay/2)

def get_guess():
    while True:
        if buttons.was_pressed(BTN_L):
            guess = 0
            break
        if buttons.was_pressed(BTN_R):
            guess = 1
            break
        if buttons.was_pressed(BTN_A):
            guess = 2
            break
        if buttons.was_pressed(BTN_B):
            guess = 3
            break
    return guess
```



New projects similar to Practice #6

Continue the Simon memory game:

- Add sounds to play with the lights
 - Use two lists – one for the lights and one for the sounds
- Use a different visual display for instructions
 - Use an arrow, image or color on the display to indicate which button to press
- Change the difficulty
 - Easy and hard could be the amount of time the sequence is displayed
 - Easy and hard could be the number of buttons used (4 and all 6)
- Add a counter
 - Keep track of the number of correct responses and display a message



Selecting a project

You should fully understand the requirements of the Create PT and are ready to start brainstorming an idea for your project. You will be given class time to complete the project. Keep these things in mind:

- Should be able to complete the code in 7 hours (not too complicated)
- Personally relevant – pick a project you are interested in
- Has a clear purpose
- No new programming skills – This isn't the time to learn something new. Stick with what you already know and make it work.



Final comments:

- Work with a partner!
 - Do equal work
 - You BOTH need to understand the code
- Your project should have a detailed intro with clear instructions
- Your project should have some kind of ending
- Start small, and build from there.
 - Get the basics done, with input and output, a list and a function.
 - Go step-by-step (have a plan)
 - Test as you go, a few lines at a time, debugging as necessary
 - Then if you have time, add more to the project.
- You can get feedback from your peers
- You can get help from your peers
- DO NOT ask the teacher!

